View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Nevada, Las Vegas Repository

Computer Science Faculty Publications

Computer Science

1-29-2019

Energy Efficiency and Renewable Energy Management with Multi-State Power-Down Systems

James Andro-Vasko
University of Nevada, Las Vegas, james.andro-vasko@unlv.edu

Wolfgang W. Bein
University of Nevada, Las Vegas, wolfgang.bein@unlv.edu

Hiro Ito
The University of Electro-Communications, itohiro@uec.ac.jp

Follow this and additional works at: https://digitalscholarship.unlv.edu/compsci_fac_articles

 Part of the Computer Sciences Commons

Repository Citation

Andro-Vasko, J., Bein, W. W., Ito, H. (2019). Energy Efficiency and Renewable Energy Management with Multi-State Power-Down Systems. *Information, 10*(2), 1-21. MDPI.
http://dx.doi.org/10.3390/info10020044

This Article is protected by copyright and/or related rights. It has been brought to you by Digital Scholarship@UNLV with permission from the rights-holder(s). You are free to use this Article in any way that is permitted by the copyright and related rights legislation that applies to your use. For other uses you need to obtain permission from the rights-holder(s) directly, unless additional rights are indicated by a Creative Commons license in the record and/or on the work itself.

This Article has been accepted for inclusion in Computer Science Faculty Publications by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact digitalscholarship@unlv.edu.

information

MDPI

Article

Energy Efficiency and Renewable Energy Management with Multi-State Power-Down Systems †

James Andro-Vasko [1], Wolfgang Bein [1,*] and Hiro Ito [2]

[1] Department of Computer Science, University of Nevada, Las Vegas, NV 89154, USA; androvas@unlv.nevada.edu

[2] School of Informatics and Engineering, The University of Electro-Communications, Tokyo 182-8585, Japan; itohiro@uec.ac.jp

* Correspondence: wolfgang.bein@unlv.edu; Tel.: +1-702-895-1477

† This paper is an extended version of our paper published in ITNG 2018.

Received: 1 December 2018; Accepted: 21 January 2019; Published: 29 January 2019

Abstract: A power-down system has an on-state, an off-state, and a finite or infinite number of intermediate states. In the off-state, the system uses no energy and in the on-state energy it is used fully. Intermediate states consume only some fraction of energy but switching back to the on-state comes at a cost. Previous work has mainly focused on asymptotic results for systems with a large number of states. In contrast, the authors study problems with a few states as well as systems with one continuous state. Such systems play a role in energy-efficiency for information technology but are especially important in the management of renewable energy. The authors analyze power-down problems in the framework of online competitive analysis as to obtain performance guarantees in the absence of reliable forecasting. In a discrete case, the authors give detailed results for the case of three and five states, which corresponds to a system with on-off states and three additional intermediate states "power save", "suspend", and "hibernate". The authors use a novel balancing technique to obtain optimally competitive solutions. With this, the authors show that the overall best competitive ratio for three-state systems is $\frac{9}{5}$ and the authors obtain optimal ratios for various five state systems. For the continuous case, the authors develop various strategies, namely linear, optimal-following, progressive and exponential. The authors show that the best competitive strategies are those that follow the offline schedule in an accelerated manner. Strategy "progressive" consistently produces competitive ratios significantly better than 2.

Keywords: online competitive analysis; energy-efficiency; power-down problems; renewable energy management

1. Introduction

With the shift towards renewable energy sources, such as biomass, wind, and solar on the power supply side and smart appliances and electric vehicles on the load side, there are enormous challenges in designing a reliable, effective and secure power infrastructures. Today, when renewables produce a surplus of energy, the surplus generally does not affect the operation of traditional power plants. Instead, renewables are throttled down or the surplus is simply ignored. In the future, it is projected that more than two-thirds of all domestic power will be generated by renewables and then this situation will not be tenable. Instead, the traditional power plants will need to be throttled down. Powering down a major power plant is not trivial and requires considerable extra cost—as noted in the authors' recent paper on managing a renewable energy infrastructure [1].

Power management and energy-efficiency are also important topics in information technology itself, given the exponential growth in the number of information technology devices and cloud

services worldwide. Finite state systems are used in electronic control, such as power optimization for laptop computers, hand-held devices, and work stations [2–5]. Eric Smith, Google's former Chief Executive Officer, remarked "What matters most to the computer designers at Google is not speed but power, low power, because data centers can consume as much energy as a city" (Quote attributed to [6]) and energy costs at Google routinely exceed hardware costs [7]. Ways to minimize energy consumption are crucial and power usage has increasingly become a first order constraint in Information Technology. There is now a body of work on algorithmic approaches for energy efficiency (e.g., see Albers et al. for a survey [8]), but it is surprising that for power-down mechanisms only a few algorithmic techniques exist. The problems the authors study in this paper have been previously addressed only in an asymptotic sense, where the number of states is large (see [9,10]). In contrast, the authors here study problems with a few states as well as systems with one continuous state. Thus, a gap remains for obtaining exact results for specific systems, especially in renewable energy management, and this study addresses this gap.

1.1. Problem Formulation

The power-down problem is formally defined as follows: consider here a system which has two states, called ON, OFF, and additionally a continuous or finite set of intermediate states. In the continuous case, the set of states is $r \in [0,1]$, where the value 0 is mapped to the ON-state; the value 1 is mapped to the OFF state; and the interval $(0,1)$ is mapped to the intermediate states. The running cost of the device in the ON state is proportional to the time of usage and the device in the OFF state consumes zero amount of energy; the intermediate states serve as sleep states, where the running cost is also proportional to time, but a smaller cost $0 < a(r) < 1$. There is no cost to switching from ON to OFF or any of the intermediate states, but a fixed cost $0 < d(r) < c$ occurs when switching from any of the intermediate states to ON, with c the cost of switching from OFF to ON. For systems with a finite number of states, instead of mapping from $[0,1]$, the states are in $\{0,\ldots,k\}$, with 0 being the ON-state, k being the OFF-state, and costs then described as a_i, d_i for $i \in \{0,\ldots,k\}$—such a system is called a $k+1$ state system. The simplest of such systems is a two-state system where there is only OFF and ON, i.e., $k = 1$.

1.2. Previous Work, Contribution, and Organization of the Paper

The authors study problems in the framework of online competitive analysis, a comprehensive introduction to such analysis can be found in [11]. In the online model, an algorithm must make decisions without any knowledge of future inputs. The authors analyze online algorithms in terms of competitiveness, a measure of performance which compares the solution obtained online with the optimal offline solution for the same problem, where the lowest possible competitiveness is best. This model has the advantage that statistical assumptions are not necessary. The existing grid has been remarkably successful and reliable, which can be attributed to the accurate modeling of load pattern on the grid to accurately predict the demand. The demand prediction allows power generation to be easily adapted to the predicted demand; see the authors' paper [1] for a survey on dependable electrical grids. However, the situation is markedly different with renewables because short-term gaps in renewable energy supply are hard to predict. As a result, the worst case analysis (i.e., not using forecasting or distributional assumptions) is more appropriate for a a truly resilient grid; worst case analysis gives a performance guarantee in the absence of reliable forecasting. For example, predictions are tenuous, as there have been extreme weather patterns in recent years, suspected by some climate scientists [12] to be related to a change in the Arctic Oscillation (OA) and North Atlantic Oscillation (NAO).

The online power-down problem with two states is equivalent to the noted "ski-rental problem", which was first studied by Karlin et al. [13] to model caching in multi-processor systems. The relevance of the ski-rental problem for power-down is mentioned in [14] as well as in the survey article on energy efficient algorithms by Albers [8]; the authors will discuss the power-down equivalence further in

Section 2. A randomized version of the problem was studied by [15] in connection with Dynamic TCP Acknowledgement.

The authors also mention that, for two-state power down problems, Bein et al. [16,17] have introduced the decrease-and-reset technique, which decreases the standby time gradually when the frequency of requests becomes low. In the decrease and reset approach, a parameter called "slackness degree" is introduced, which represents the frequency of arrivals; a near optimal algorithm can be constructed, which has superior competitiveness in the presence of slackness. It is noted that this approach is reliant on statistical assumptions and the authors do not pursue this approach in this paper.

Online multi-state systems were first studied in an asymptotic sense for systems with a large number of states by Augustine, Irani and Swami in [9,10]. They gave an algorithm to produce an approximation strategy and also a simple algorithm that achieves a competitive ratio of $3 + 2\sqrt{2} \approx 5.83$. There is much related work on speed scaling of processors [8]. This technique saves energy by throttling down the speed of a processor whenever possible, e.g., [18–20]. Chen et al. [21] consider the problem of online dynamic power management which provides hard real-time guarantees for multi-processor systems and Albers et al. [22] give multi-processor speed scaling with migration. This is part of a large body of work on scheduling jobs online on parallel machines with hard deadlines, see [23] for recent results. Other research considered power-down problems over a network to reduce the energy cost of idling server machines while maintaining an effective network [2–5]. This is part of a larger theme of using power management to achieve energy savings in data centers; see the authors' recent article [24] as well as [25].

The work by Augustine et al. [9,10] is closest to the authors' work, but here the focus is on systems with a small number of finite states as well as systems with one continuous state. Such systems play a role in energy efficiency and renewable energy management. The purpose of this paper is to fill the gap between asymptotic techniques and exact solutions for systems with states. The authors present here a novel balancing technique, and note that these techniques are markedly different from those in previous work. The authors obtain optimal and closed form solutions and report new results for numerous power-down systems. It is known that a simple strategy for the ski-rental problem yields a competitive ratio of 2 (see e.g., [13] or textbook [11]). The authors show that the overall best competitive ratio for three-state systems is $\frac{9}{5}$ and they obtain optimal ratios for various five state systems. For the continuous case, the authors develop various strategies, namely linear, optimal-following, progressive and exponential. The authors show that best competitive strategies are those that follow the offline schedule in an accelerated manner; strategy "progressive" consistently produces competitive ratios significantly better than 2.

The paper is organized as follows: in Section 2, the authors provide details of the online model, and show its relevance to energy efficiency and renewable energy management. The authors continue in Section 3 to analyze three-state systems and generalize such analysis in Section 4 for the situation of five states. Five state systems are inspired by devices used in everyday life, where there is the ON and OFF state and then three intermediate states "power save", "suspend", and "hibernate". Based on these techniques in that section, the authors also outline a heuristic which works well for more than five states. In Section 5, the authors present the continuous model, and focus on five online power-down strategies: linear, optimal-following, progressive, logarithmic and exponential. Section 6 presents the conclusions.

2. Online Computation and the Smart Grid

In simulation and modeling, frequently statistical or distributional assumptions are made; for the power-down problem, models make assumptions about the distribution of requests. For example, in the traditional power power grid, the demand for power can be predicted throughout the day, and across the seasons, power demand can be predicted with a certain degree of certainty [26]. With renewable and decentralized power supplies, such predictions are much more precarious. Here, the competitive ratio is useful as a guarantee no matter how unusual the situation becomes.

Online algorithms make decisions without knowledge of future input. For the power-down problem, this means that an online algorithm will decide at each moment how to switch states without knowing what the next request will be. In contrast, an offline algorithm can make decisions based on the full knowledge of the entire input sequence. For example, algorithm A has competitive ratio C for a given request sequence σ, if

$$Cost_A(\sigma) \leq c \cdot Cost_{opt}(\sigma) \tag{1}$$

with $Cost_A(\sigma)$ the cost of A to serve σ, and $Cost_{opt}(\sigma)$ the cost of the optimal offline algorithm on σ. If this is true for all feasible input sequences, then we say that the online algorithm is C-competitive. The competitive ratio for a given algorithm A is defined as the smallest such C possible. It is desirable to find the the online algorithm with the smallest competitive ratio, which is also referred to as the competitive ratio of the problem. See [11] for a broad introduction to the theory of online algorithms.

Returning now to the formulation of the power-down problem, at any time, the device may be in any state but it must be turned to the ON state when service is requested. Let $t_1^s, \ldots, t_n^s$ and $t_1^e, \ldots, t_n^e$ be non-negative real values that represent requests for service between start of service times t_i^s and end of the service times t_i^e, $(i = 1, 2, ..., n)$. Note that $0 \leq t_1^s < t_1^e < t_2^s < t_2^e < \cdots < t_n^s < t_n^e$ holds. Thus, at time t_i^s, the state of a device must be in ON until time t_i^e. In between requests the device can remain in the ON state or go to the OFF state or any of the intermediate states. Note that if t_i^e is very close to t_{i+1}^s it may be inefficient to turn the device off. Instead it could be advantageous to keep the machine on or perhaps operate the device in any of the intermediate states. It is important that during usage, the device must be ON—both offline and online. Thus, under competitive analysis the length of the service $t_i^e - t_i^s$ is irrelevant—the issue is only whether the machine switches to a new state at time t_i^e—and one can thus assume without loss of generality, those usage times of the device are infinitesimal. Therefore, it is necessary to redefine the input sequence as where $t_i := t_i^s = t_i^e$ and define a request sequence in terms of the arrival time of request i.

Under competitive analysis, for the two-state problem, an optimal online algorithm, here referred to as $\mathcal{S}$, is well known for this problem [13]: $\mathcal{S}$ switches to ON at a service request. After the service, $\mathcal{S}$ remains in ON to stand by for time c units—recall that c is the cost of switching from OFF to ON. This means that if another service is requested within the standby period, $\mathcal{S}$ simply remains in state ON. Only if the waiting time is larger than c will $\mathcal{S}$ go to OFF after a standby time of c. Algorithm $\mathcal{S}$ is 2-competitive, which is optimal.

This situation is analogous to the noted ski rental problem [13]. In the ski rental problem, a "friend" invites an invitee to go skiing, where one can rent skis at cost \$1 or purchase them at cost $\$c = m \cdot \$1$, for some $m \in \mathbb{N}$. Unlike the inviter, the invitee does not know how often he/she will be invited. Thus, the invitee has to decide how often to rent and when to buy. It is easy to see that the best competitive ratio of the invitee cost versus the inviter cost achievable by any online algorithm for the ski rental problem is 2; the strategy is to rent m times before buying. The ski rental problem is analogous to the two-state power-down problem in the following way; both problems are about when to commit to a higher cost in anticipation of future savings. In the ski-rental problem, this commitment is when buying skis, whereas, in the power-down problem, the commitment is to accept a (large one-time) switching-on cost when the system is next switched off.

3. Systems with Three States

In the three-state problem, the states are ON, OFF, and one intermediate state, or INT. In the definition of power-down systems, this means $k = 2$, i.e., the states are $\{0, 1, 2\}$ named {ON, INT, OFF}. For convenience, the authors assume that both the running cost in state ON, as well as the switching cost from OFF to ON are 1. Furthermore, the costs of state INT are referred to as a and d instead of a_1 and d_1, since the index is redundant when there is only one intermediate state. The costs of the system are summarized in the Table 1 below:

Table 1. A three-state power-down system with running cost a in the intermediate state INT and switching cost d to switch from INT to ON.

State	Running Cost	Switching Cost
ON	1	0
INT	$a \in (0,1)$	$d \in (0,1)$
OFF	0	1

An online algorithm for this problems is fully described by specifying two switching times x_1 and x_2 which denote the switch times from state ON to state INT and then state INT to state OFF. In contrast, the optimal offline algorithm $\mathcal{OPT}$ is described by two values x_{opt_1} and x_{opt_2} as follows: if the request arrives before x_{opt_1}, then $\mathcal{OPT}$ will be in the ON state during the idle duration; if the request arrives between x_{opt_1} and x_{opt_2}, then $\mathcal{OPT}$ will be in the INT state, and if the request arrives after x_{opt_2}, then $\mathcal{OPT}$ will be in the OFF state. Thus, Cost_{opt} is defined as follows:

$$\text{Cost}_{opt}(t) = \begin{cases} t & \text{if } t < x_{opt_1}, \\ at + d & \text{if } x_{opt_1} \leq t < x_{opt_2}, \\ 1 & \text{if } t \geq x_{opt_2}, \end{cases} \tag{2}$$

where the values of x_{opt_1} and x_{opt_2} for given a and d are according to the following theorem:

Theorem 1. *$\mathcal{OPT}$ assigns $x_{opt_1} = \min\{d/(1-a), 1\}$ and $x_{opt_2} = \max\{(1-d)/a, 1\}$.*

Proof. The offline cost curves are $f(t) = t$, $f(t) = at + d$, and $f(t) = 1$. The curve $f(t) = t$ intersects with $f(t) = at + d$ when $t = d/(1-a)$; before this time, the ON state yields the optimal cost and after this time, the INT state yields the optimal cost. The curve $f(t) = at + d$ intersects with $f(t) = 1$ at $t = (1-d)/a$. If the request arrives before $t = d/(1-a)$, then from 0 to $d/(1-a)$, the optimal cost is obtained using the cost curve $f(t) = t$ which is the ON state; if the request arrives between $d/(1-a)$ to $(1-d)/a$, then the optimal cost is obtained using the cost curve $f(t) = at + d$, which is the INT state, and if the request arrives at or after $(1-d)/a$, the optimal cost curve is $f(t) = 1$, which is the OFF state. Therefore, $x_{opt_1} = d/(1-a)$ and $x_{opt_2} = (1-d)/a$.

If $a + d \geq 1$, it is easily verified that it is not advantageous to use INT. Therefore, the lemma follows. □

Recall that any online algorithm starts in the ON state, at some point switches to INT and then to OFF state as given by values for x_1 and x_2. The cost of such an algorithm, $Cost_{online}$, is

$$\text{Cost}_{\text{online}}(t) = \begin{cases} t & \text{if } t < x_1, \\ x_1 + a(t - x_1) + d & \text{if } x_1 \leq t < x_2, \\ x_1 + a(x_2 - x_1) + 1 & \text{if } t \geq x_2. \end{cases} \tag{3}$$

For such online algorithms, it is necessary to determine its x_1 and x_2 values as to minimize its competitive ratio.

Lemma 1. *For an online algorithm with minimal competitiveness, it is necessary that (a) $x_1 \leq x_{opt_1}$ and (b) $x_2 = x_{opt_2}$.*

Proof. For (a), it is assumed that $x_1 > x_{opt_1}$. If this is the case, $x_{opt_1} = x_1 + \delta$ such that $\delta > 0$. Then, the online algorithm stays in ON for $x_1 + \delta$, whereas the offline algorithm would have chosen to be

in the intermediate state, thus incurring only a cost of $a(x_{opt_1} + \delta)$. Therefore, there is the following competitive ratio:

$$\frac{x_{opt_1} + \delta + d}{a(x_{opt_1} + \delta) + d}. \tag{4}$$

The online and offline costs can be compared to get:

$$x_{opt_1} + \delta + d \geq a(x_{opt_1} + \delta) + d, \tag{5}$$

$$x_{opt_1} + \delta \geq a(x_{opt_1} + \delta). \tag{6}$$

It is clear that as $\delta > 0$ increases the competitive ratio increases, since the online costs increase at a faster rate than the offline cost.

To prove (b) again, assume the contrary, namely that $x_2 < x_{opt_1}$, and therefore $x_2 = x_{opt_1} - \delta$ where $\delta > 0$. In this case, the online algorithm goes from ON to INT to OFF, whereas the offline algorithm remains in ON. Quantifying this behavior, the competitive ratio is:

$$\frac{x_1 + a(x_{opt_1} - \delta - x_1) + 1}{x_{opt_1} - \delta} \tag{7}$$

or

$$a + \frac{x_1(1-a) + 1}{x_{opt_1} - \delta} \tag{8}$$

As δ increases, the competitive ratio increases as well and $x_2 \geq x_{opt_1} \geq x_1$. So now to show that $x_2 = x_{opt_2}$ must be true, one first assumes $x_2 > x_{opt_2}$, so the competitive ratio would be $x_1 + a(x_2 + \delta - x_1) + 1$. It is clear that as δ increases, the competitive ratio will increase linearly. Now assume to examine the competitive ratio if $x_{opt_1} \leq x_2 < x_{opt_2}$. So $x_2 = x_{opt_2} - \delta$. The competitive ratio would be:

$$\frac{x_1 + a(x_{opt_2} - \delta - x_1) + 1}{a(x_{opt_2} - \delta) + d} \tag{9}$$

or

$$1 + \frac{x_1(1-a) + 1 - d}{a(x_{opt_2} - \delta) + d}. \tag{10}$$

Once again, as δ increases, the competitive ratio is increases. Thus, when $x_2 > x_{opt_2}$ and $x_1 \leq x_2 < x_{opt_2}$, both lead to contradictions because the competitive ratio will not be minimal in those cases; thus, $x_2 = x_{opt_2}$ to minimize the competitive ratio. □

From Lemma 1, the competitive ratio for a three-state system depends only on the value of x_1, since $x_2 = x_{opt_2}$ and x_{opt_2} is fully determined by the values of a and d. Clearly, the maximum ratio between $\mathcal{OPT}$ and $\mathcal{A}$ occurs when $\mathcal{A}$ has just changed states i.e., at x_1 and x_2. Define CR_1 and CR_2 as the competitive ratios at x_1 and x_2; then CR_1 and CR_2 can be written as:

$$CR_1 = \frac{x_1 + d}{x_1}, \tag{11}$$

$$CR_2 = x_1 + a(x_2 - x_1) + 1. \tag{12}$$

This is because the ratios at the two breaking points have to match. The goal is to minimize the worst case competitive ratio at the switching time values. The competitive ratio of the system will be $\max\{CR_1, CR_2\}$. We have:

Lemma 2. *The competitive ratio for the three-state system is minimized when $CR_1 = CR_2$.*

Proof. First, assume $CR_1 < CR_2$ and the competitive ratio is minimal. If now the value for x_1 would be decreased by an arbitrarily small constant such that $CR_1 < CR_2$ is still preserved, the value for CR_1 would increase, but the value for CR_2 would decrease from (11) and (12), respectively. This leads to a contradiction because the competitive ratio was not minimal. If $CR_1 > CR_2$, the value of x_1 can be increased while still maintaining $CR_1 > CR_2$. This also leads to a contradiction, since, with the competitive ratio between CR_1 and CR_2, the maximum of the two has decreased. □

For a three-state system, the values of a and d can be any value between 0 and 1, and, for convenience, set $\lambda = a + d$. From Lemma 2 to obtain the optimal competitive ratio, it is known that $CR_1 = CR_2$ must be true. From Lemma 1, the value of x_2 is known, and setting $CR_1 = CR_2$ is used to obtain the optimal x_1 value as follows:

$$\frac{x_1 + d}{x_1} = x_1 + a(x_2 - x_1) + 1. \tag{13}$$

Solving for x_1, we obtain:

$$x_1 = \frac{ax_2 - \sqrt{4d - 4ad + a^2x_2^2}}{2(a-1)} \tag{14}$$

$$= \frac{(\lambda - d)x_2 - \sqrt{4d - 4d(\lambda - d) + x_2^2(\lambda - d)^2}}{2(\lambda - d - 1)}. \tag{15}$$

Using the value of x_1, it is possible to substitute Equation (14), into CR_1 or CR_2 and that will yield the optimal competitive ratio given a value for a and d:

$$CR_{opt} = 1 + \frac{2d(\lambda - d - 1)}{(\lambda - d)x_2 - \sqrt{4d - 4d(\lambda - d) + x_2^2(\lambda - d)^2}}. \tag{16}$$

Using Equation (16), it is possible to give a value for λ, and search for the optimal a and d values that will minimize the competitive ratios. We obtain:

Theorem 2. *For a three-state system where $a + d = 1$, the best competitive ratio achievable is $CR = \frac{9}{5}$ when $a = \frac{3}{5}$ and $d = \frac{2}{5}$.*

Proof. Consider Equation (16) and compute the derivative with respect to d:

$$\left(\frac{\delta\, CR_{opt}}{\delta d}\right) = \frac{10d - 2}{4\sqrt{5d^2 - 2d + 1}} - \frac{1}{2} = 0. \tag{17}$$

After simplifying the above expression, we get

$$20d^2 + 8d = 0. \tag{18}$$

Solving for d, we obtain $d = \frac{2}{5}$, and $a = \frac{3}{5}$, and with those values the competitive ratio is $\frac{9}{5}$. □

In Tables 2 and 3, the minimum competitive ratio for various λ values, where $\lambda = a + d$ is tabulated.

Table 2. Best competitive ratios (CR) together with the corresponding online algorithm for that ratio (defined by switching time values x_1 and x_2) for given parameter value λ.

a	d	λ	CR	x_1	x_2
0.0512	0.0488	0.1	1.9976	0.0489	18.580
0.1046	0.0954	0.2	1.9908	0.0963	8.6487
0.1600	0.1400	0.3	1.9800	0.1429	5.3750
0.2173	0.1827	0.4	1.9654	0.1893	3.7613
0.2764	0.2236	0.5	1.9472	0.2361	2.8090
0.3373	0.2627	0.6	1.9254	0.2839	2.1859
0.4000	0.3000	0.7	1.9000	0.3333	1.7500
0.4646	0.3354	0.8	1.8708	0.3852	1.4305
0.5312	0.3688	0.9	1.8376	0.4403	1.1883
0.6000	0.4000	1.0	1.8000	0.5000	1.0000
0.6312	0.4688	1.1	1.8376	0.5597	1.0000
0.6646	0.5354	1.2	1.8708	0.6148	1.0000
0.7000	0.6000	1.3	1.9000	0.6667	1.0000
0.7373	0.6627	1.4	1.9254	0.7161	1.0000
0.7764	0.7236	1.5	1.9472	0.7639	1.0000
0.8172	0.7827	1.6	1.9654	0.8107	1.0000
0.8600	0.8400	1.7	1.9800	0.8571	1.0000
0.9046	0.8954	1.8	1.9908	0.9037	1.0000

Table 3. Best competitive ratios (CR) together with the corresponding online algorithm for that ratio (defined by switching time values x_1 and x_2) for various λ values close to 1.

a	d	λ	CR	x_1	x_2
0.565305	0.384695	0.95	1.81939	0.46949	1.088447829
0.572196	0.387804	0.96	1.81561	0.475479	1.0699061161
0.579111	0.390889	0.97	1.81178	0.481522	1.0518035403
0.58605	0.39395	0.98	1.8079	0.487622	1.034126781
0.593012	0.396988	0.99	1.80398	0.493781	1.0168630652
0.6	0.4	1	1.8	0.5	1
0.603012	0.406988	1.01	1.80398	0.50622	1
0.60605	0.41395	1.02	1.8079	0.512377	1
0.609111	0.420889	1.03	1.81178	0.518478	1
0.612196	0.427804	1.04	1.81561	0.524522	1
0.615305	0.434695	1.05	1.81939	0.530511	1

In Tables 2 and 3 and Figure 1, if the optimal competitive ratios are found for several λ values; the overall optimal competitive ratio is obtained when $\lambda = 1$. Based on those experimental results, one can see that for a three-state machine, the competitive ratio is optimal when $a + d = 1$. Such a system is called and "ideal system". In practice, therefore, it is advantageous to choose a system with $a + d = 1$ whenever possible, as in this case there is a power-down strategy with optimal competitive ratio of 1.8. If, for example, there is a degree of freedom in designing an intermediate state for a plant then the intermediate state should ideally be such that $a + d = 1$ holds.

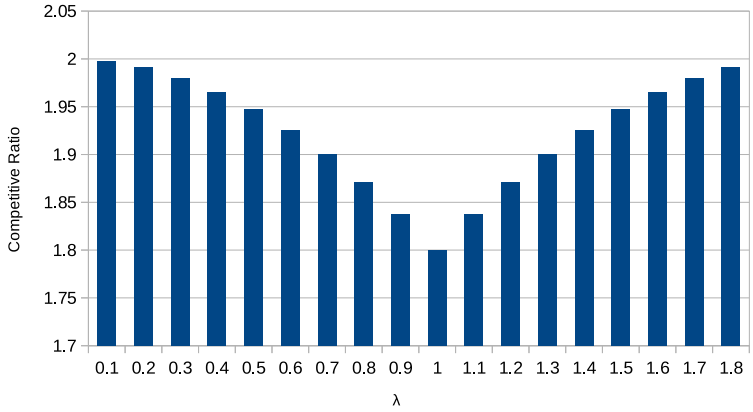

Figure 1. Overall optimal competitive ratios for given λ values.

4. Systems with Few States

In this section, the authors consider systems with more than three-states, specifically with an arbitrary but small number of states. Generalizing from the three-state situation, for a $k+1$-state situation, $\mathcal{OPT}$ is fully described by values $x_{opt_1}, \ldots, x_{opt_k}$ and the online Algorithm $\mathcal{A}_k$ is fully described by values $x_1, \ldots, x_k$. Noting that the lemmas can be generalized inductively to a system with more than three-states, these are stated without proof:

Lemma 3. *For an online algorithms with minimal competitiveness, it is necessary that (a)* $x_1 \leq x_{opt_1}, \ldots, x_{k-1} \leq x_{opt_{k-1}}$ *and (b)* $x_k = x_{opt_k}$.

Lemma 4. *For a* $k+1$ *state power-down system, the maximal competitive ratio appears at a transition, i.e., at a value in* $\{x_1, \ldots x_k\}$.

Lemma 5. *Given an online algorithm for a* $k+1$ *state power down problem with minimal competitive ratio CR. Then,* $CR = CR(x_1) = \cdots = CR(x_k)$.

Based on these lemmas, the authors present a heuristic for finding an optimally competitive algorithm. As in the three-state case, one can write the competitive ratio out at all transition points and use the fact from Lemma 5 that, in order for CR to be minimal, it must hold that $CR(x_1) = \cdots = CR(x_k) = CR$, which yields the following system system of equations:

$$\frac{x_1 + d_1}{x_1} = CR, \tag{19}$$

$$\frac{x_1 + a_1(x_2 - x_1) + d_2}{\min\{x_2, a_1 x_2 + d_1\}} = CR, \tag{20}$$

$$\frac{x_1 + a_1(x_2 - x_1) + a_2(x_3 - x_2) + d_3}{\min\{x_3, a_1 x_3 + d_1, a_2 x_3 + d_2\}} = CR. \tag{21}$$

$$\vdots$$

In the previous equations, the numerator reflects the fact that the online algorithms shifts down from state to state, whereas the denominator comes from the best choice available to the offline algorithm. Furthermore, as in the three-state case, these equations reflect the requirement that the ratios at the breaking points have to match.

The authors' heuristic, which is detailed below, searches competitive ratios CR; for each CR value considered, the corresponding x_i values can be obtained by solving for x_i in the previous system. It is

shown in [10] that there exists a solution which has competitive ratio $3+2\sqrt{2}$ and this is taken as an initial solution. Next, given a desired precision δ, a competitive ratio search in the range $[1, 3+2\sqrt{2}]$ is performed. This is guided by the following: if $x_k > x_{opt_4} + \theta$, the competitive ratio CR can only be minimal if $x_4 = x_{opt_4}$ due to Theorem 3; thus, CR must be reduced. If $x_4 < x_{opt_4}$, then there there is no CR-competitive solution and CR must be increased. This process continues until $x_{opt_4} \geq x_4 \leq x_{opt_4} + \delta$; a ratio arbitrarily close to optimal is obtained.

Algorithm 1 was used to model a typical system with few states, namely a quintuple system. Such a system is inspired by devices used in everyday life, where there is the ON and OFF state, and three intermediate states "power save", "suspend", and "hibernate". The authors have simulated a system with running costs $a_0 = 1, a_1 = 0.55$, $a_2 = 0.4$, $a_3 = 0.25$ $a_4 = 0$ and switching costs $d_0 = 0$, $d_1 = 0.225$, $d_2 = 0.4$, $d_3 = 0.60$, $d_4 = 1$. The iterations of the simulation using Algorithm 1 are shown in Table 4.

Algorithm 1 Power-down heuristic

Given values $a_{0\to k}$, and $d_{0\to k}$
lowerBound = 1, upperBound = 3 + $2\sqrt{2}$;
CR = (lowerBound + upperBound) / 2;
Compute $x_1, \ldots, x_k$ using CR;
while $x_k < x_{opt_k}$ *or* $x_k > x_{opt_k} + \delta$ **do**
  if $x_k < x_{opt_k}$ **then**
    lowerBound = CR;
  else
    upperBound = CR;
  end
  CR = (lowerBound + upperBound) / 2;
  Recalculate $x_1, \ldots, x_4$;
end

Table 4. Sample run of Algorithm 1: for each iteration, the current competitive ratio (CR) together with the the corresponding online algorithm for that ratio (defined by switching time values x_i) is shown.

Iteration	x_1	x_2	x_3	x_4	lowerBound	upperBound	CR
1	0.0932	0.1543	0.2207	9.2632	1.000	5.828	3.414
2	0.1864	0.2920	0.4027	4.0756	1.000	3.414	2.207
3	0.3731	0.6249	1.0401	0.7414	1.000	2.207	1.603
4	0.2486	0.3778	0.5248	2.6309	1.603	2.207	1.905
5	0.2984	0.4438	0.7193	1.7810	1.603	1.905	1.754
6	0.3314	0.4864	0.8488	1.3184	1.603	1.754	1.679
7	0.3142	0.4643	0.7815	1.5509	1.679	1.754	1.716
8	0.3061	0.4538	0.7496	1.6670	1.716	1.754	1.735
9	0.3099	0.4587	0.7645	1.6122	1.716	1.735	1.726
10	0.3121	0.4615	0.7729	1.5816	1.716	1.726	1.721
11	0.3108	0.4598	0.7678	1.6000	1.721	1.726	1.724

Based on the value of x_4, at each iteration, a new value for the competitive ratio between the upper bound and the lower bound is selected. The authors note that using Theorem 5 which implies $CR(x_1) = CR(x_2) = CR(x_3) = CR(x_4)$ the other x_i values follow. Thus, a binary search on the competitive ratio in the range $[1, 3+2\sqrt{2}]$ is performed, until a value such that $x_4 \equiv_{\text{mod } \delta} x_{opt_4}$ is

achieved. Figure 2 displays the situation from Table 4. Ratios are maximized at transition times, decreasing as the standby time moves away from any transition time until the next transition time is reached.

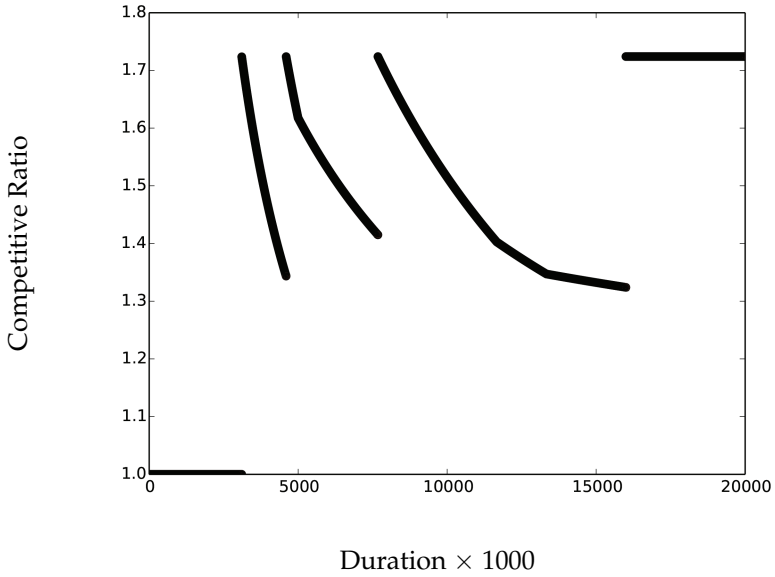

Figure 2. The solution obtained in the sample run of Table 4. Per Lemma 4 at switching times, all ratios are equal.

The authors have conducted extensive simulations for various quintuple systems. Results are shown in Table 5.

Table 5. Simulations results for various five-state systems (given by a_i and d_i parameter values) when $\theta = 0.01$. Competitive ratios are well below 2.

i	a_i	d_i	x_i	CR	a_i	d_i	x_i	CR
0	1.0000	0.0000	0.0000		0.0000	1.0000	0.0000	
1	0.7500	0.2500	0.3566		0.6000	0.2000	0.2706	
2	0.5000	0.5000	0.6195	1.701	0.4000	0.4000	0.4462	1.739
3	0.2500	0.7500	0.8277		0.2000	0.6000	0.6990	
4	0.0000	1.0000	1.0001		0.0000	1.0000	2.0086	
i	a_i	d_i	x_i	CR	a_i	d_i	x_i	CR
0	1.0000	0.0000	0.0000		1.0000	0.0000	0.0000	
1	0.6000	0.1000	0.1290		0.7000	0.2000	0.2614	
2	0.4000	0.3000	0.3744	1.775	0.3000	0.4000	0.4492	1.765
3	0.1000	0.6000	0.8256		0.1000	0.8000	1.5343	
4	0.0000	1.0000	4.0083		0.0000	1.0000	2.0001	
i	a_i	d_i	x_i	CR	a_i	d_i	x_i	CR
0	1.0000	0.0000	0.0000		1.0000	0.0000	0.0000	
1	0.8000	0.1000	0.1376		0.5500	0.2250	0.3108	
2	0.5000	0.4000	0.4614	1.7265	0.4000	0.4000	0.4598	1.724
3	0.1000	0.8000	0.9003		0.2500	0.6000	0.7678	
4	0.0000	1.0000	2.0043		0.0000	1.0000	1.6000	

The authors note that the heuristic described here does not rely on asymptotic assumptions on a large number of states, in fact the heuristic is faster when there are few states as only few $x_1, \ldots, x_k$ must be recalculated in each step. The heuristic is thus tailored to the situation of few states, unlike a meta-heuristic, or the technique in [9,10].

5. Continuous Models

5.1. Offline and Online Strategies

The authors now turn to the power-down problem with a continuous number of states. Such systems are quite useful in practice: power-down systems either feature a few intermediate states, up to five, perhaps, or a continuous number of states. The following model is used here: running costs are of the form $a(r) = 1 - r^a$ and switching costs are modeled as $d(r) = cr^d$, $r \in [0, 1]$ with given parameters $a, d, c > 0$. Figure 3 shows the model for $a = 3$, $d = 5$, $c = 1.5$. The techniques developed in this section can be generalized to various other models with continuous cost functions. In this section, offline and online strategies are conducted for these models.

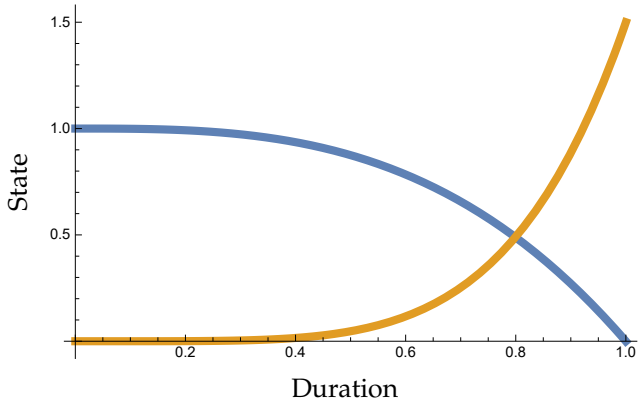

Figure 3. $a(r)$ (Brown) and $d(r)$ (Blue) curves.

It is important to remember that the behavior of the offline algorithm has full clairvoyance, and thus has full knowledge of future idle times. With this clairvoyance, one can make use of the derivative of $a(r) + d(r)$ and derive $\text{Stategy}_{\text{OFF}}(r) = (\frac{a \cdot r}{d \cdot c})^{\frac{1}{d-a}}$. Seeking to minimize the cost from the start of the idle period to its end one, $\text{Stategy}_{\text{OFF}}(r)$ is obtained as the optimal cost for this system. From this, the threshold value, τ, is calculated, which denotes the time when the machine powers down completely. In the next Equation (22), the $\text{Stategy}_{\text{OFF}}(r)$ is set to the value 1, which denotes the state of the machine when it is fully powered off:

$$\left(\frac{a \cdot \tau}{d \cdot c}\right)^{\frac{1}{d-a}} = 1. \tag{22}$$

By setting $\text{Stategy}_{\text{OFF}}(r)$ to this state value, one can simply solve (22) to determine the τ value at which the machine is powered off; one obtains:

$$\tau = \frac{c \cdot d}{a}. \tag{23}$$

The authors now turn to online strategies. Given a strategy $\text{Strategy}_{\text{ONLINE}}(r)$, there is an online cost

$$\text{Cost}_{\text{ONLINE}}(r) = \int_0^r a(\text{Strategy}_{\text{ONLINE}}(r))dr + d(\text{Strategy}_{\text{ONLINE}}(r)), \tag{24}$$

where the first term describes the accumulated run time as the online algorithms moves along states and the second term described the switching cost from the resulting state. As before in the discrete case, note that, unlike an offline algorithm, an online algorithm moves from an initial state to lower power states before the next request.

It is noted that the offline cost is simply given by the cost to be in the best state for the situation (first term) plus the switching cost (second term):

$$\text{Cost}_{\text{OFF}}(r) = r \cdot a(\text{Strategy}_{\text{OFF}}(r)) + d(\text{Strategy}_{\text{OFF}}(r)). \tag{25}$$

Generalizing the ski-rental strategy to the continuous case, one can define the "Lower Envelope" approach [10] where the online algorithms mimics offline behavior; thus, a first continuous version of the lower envelope strategy is given:

$$\text{Strategy}_{\text{ONLINE}}(r) = \text{Strategy}_{\text{OFF}}(r). \tag{26}$$

In Figures 4 and 5, it can be seen that the competitive ratio increases as idle time passes. When this reaches the value of τ, the competitive ratio is the largest, with a competitive ratio of 2.

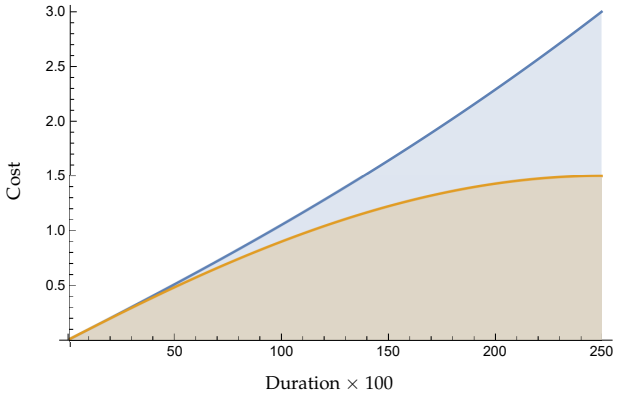

Figure 4. Cost of OFFLINE (Brown) and ONLINE (Blue).

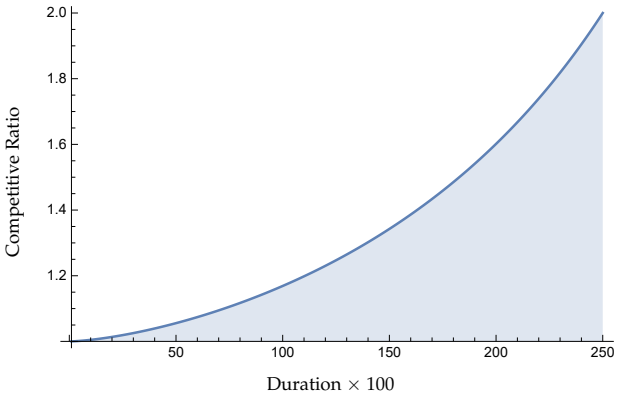

Figure 5. Competitive ratio.

A key question is to explore the strategies that can improve this canonical competitiveness of 2. In other words, what are "better-than-2-competitive" strategies? To this end, the authors have investigated various online strategies. The primary idea is to decelerate or accelerate the online strategy as compared to offline. The strategies below are referred to as "linear", "logarithmic", "exponential", and progressive due to their individual characteristics; in the definitions given, the values C, t, and z are control parameters to tune acceleration or deceleration of the various online strategies:

$$\text{Linear}(r) = r/\tau, \tag{27}$$

$$\text{Logarithmic}(C, r) = \ln(Cr)/\ln(C\tau + 1), \tag{28}$$

$$\text{Exp}(C, r) = (e^{Cr} - 1)/(e^{C\tau} - 1), \tag{29}$$

$$\text{Progressive}_{t,z}(r) = \text{Strategy}_{\text{OFF}}(r)(1 + r^{t+z} - r^{1+z}). \tag{30}$$

The authors show the strategies in Figure 6 and note that both the depicted linear and exponential strategies are below optimal offline which means that progress to lower power states is at a slower rate. Alternatively, logarithmic and progressive lie above the optimal offline; thereby, it is observed that the transition to lower power states is at faster rates than optimal offline.

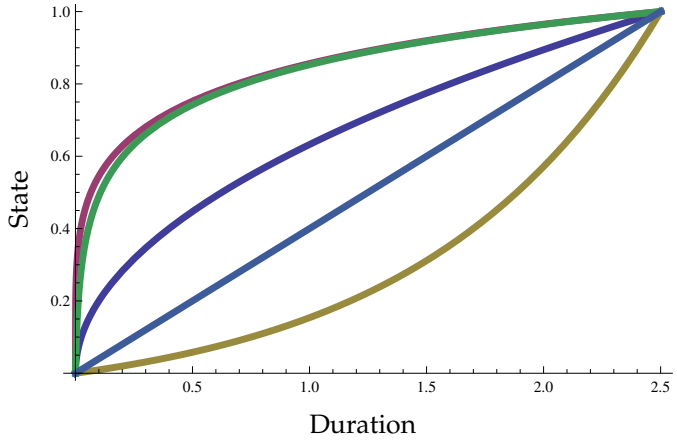

Figure 6. $\text{Exp}(1, r)$ (Brown), Linear (Blue), $\text{Strategy}_{\text{OFF}}$ (Dark Blue), $\text{Logarithmic}(200, r)$ (Green), $\text{Progressive}_{0.312,0.1}(r)$ (Violet) strategy curves.

5.2. A Summary of Simulation Results

In this subsection, the authors give simulations results to gain insight into how these strategies perform with regards to competitiveness.

This starts with strategies that transition to lower states less aggressively: these are linear and exponential, see Figures 7 and 8; and then looks to strategies with more aggressive transitioning, as in logarithmic, see Figure 9. For short idle periods, both linear and exponential show lower costs when juxtaposed to logarithmic. However, in the case of longer idle periods, when a request is similar to τ, logarithmic is best. It is clear that both linear and exponential would be sub-par here as these strategies remain in a higher power state for a longer stretch, resulting in wasted energy.

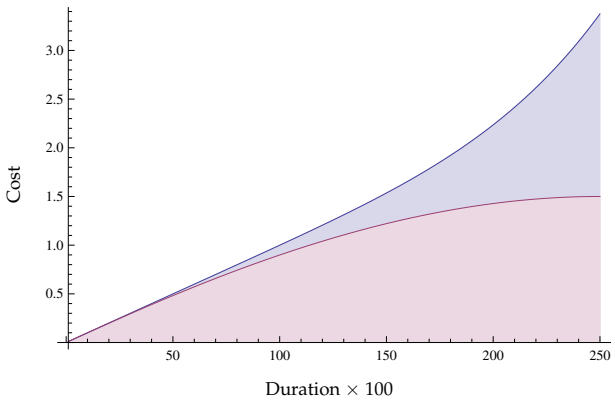

Figure 7. Cost Linear(r) (Violet) and Cost OFFLINE (Purple).

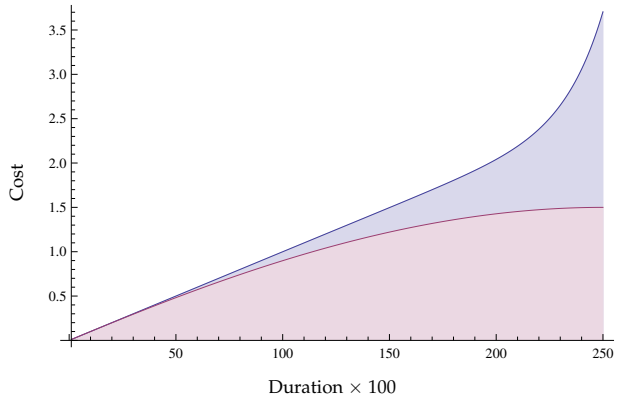

Figure 8. Cost Strategy$_e(1, r)$ (Violet) and Cost OFFLINE (Purple).

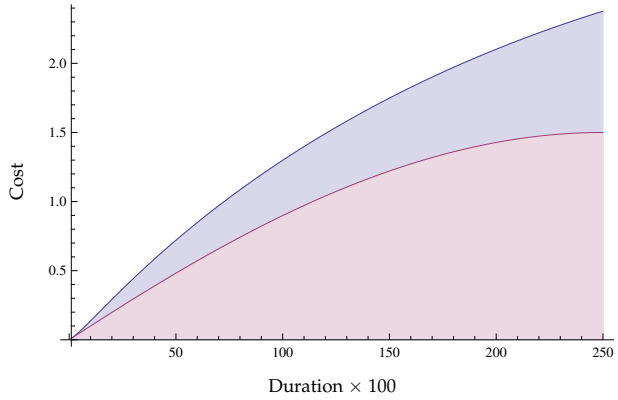

Figure 9. Cost Strategy$_{\ln}(200, r)$ (Violet) and Cost OFFLINE (Purple).

Competitive ratios for linear, exponential and logarithmic are depicted in Figures 10–12. Indeed, the competitive ratios validate what was noted above. Linear and exponential perform better for the beginning of the idle time compared to logarithmic, but soon they get worse as the idle duration increases. Thus, it is advantageous overall to power-down to lower power states at a faster rate than the offline curve.

Next is a host of examples for the progressive strategy.

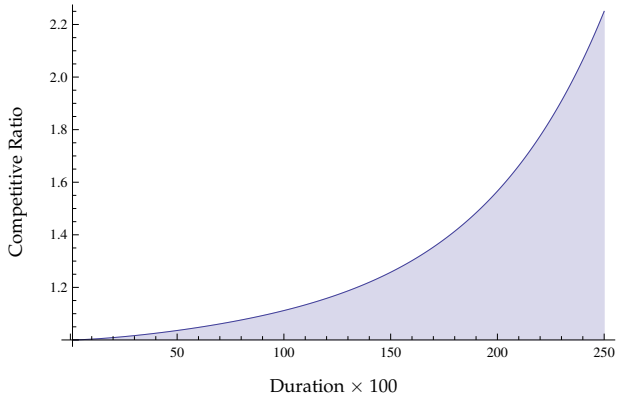

Figure 10. Competitive ratio linear function.

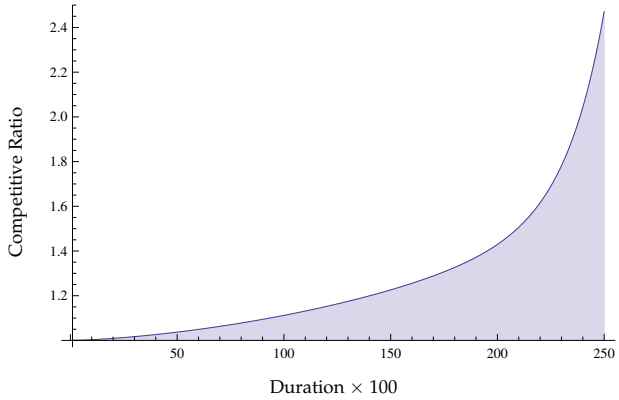

Figure 11. Competitive ratio strategy$_e(1, r)$.

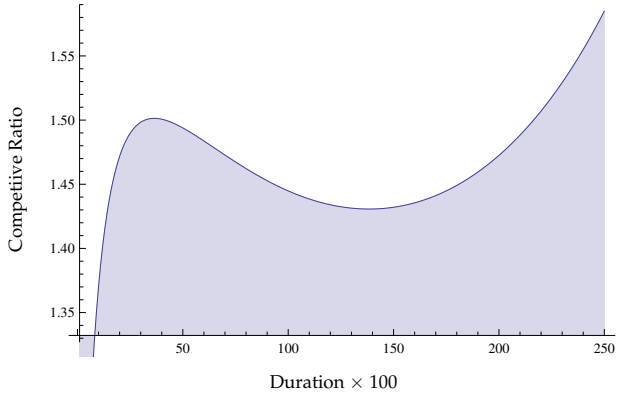

Figure 12. Competitive ratio strategy$_{\ln}(200, r)$.

In Figures 13–15, Progressive$_{t,z}(r)$ for various parameter values t and z are given. Note that, if t and z are not too large, the online strategy goes to a lower power state more swiftly. Note that clearly the control parameters t and z can be used to control the rate at which the online strategy Progressive$_{t,z}(r)$ reaches smaller power states. Additionally, t causes a more significant impact on the function than the value of z since one can increased t by only a smaller amount than to have a dramatic effect. In the experimentation, the competitive ratios for a range of values t and z can be observed.

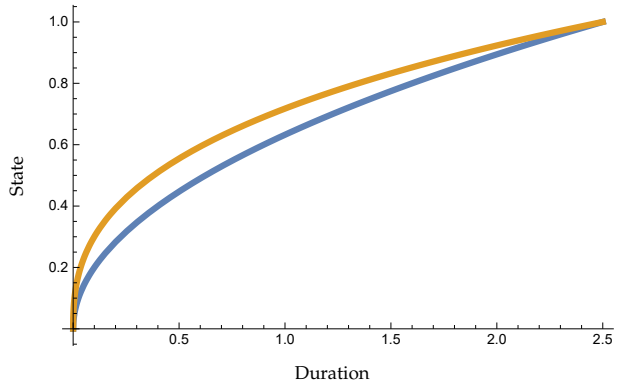

Figure 13. Strategy OPT (Blue) and online strategies for $t = 0.712$, $z = 0.1$ (Brown).

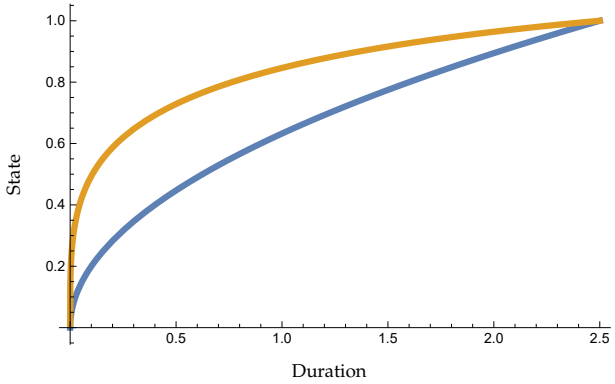

Figure 14. Strategy OPT (Blue) and online strategies for $t = 0.312$, $z = 0.2$ (Brown).

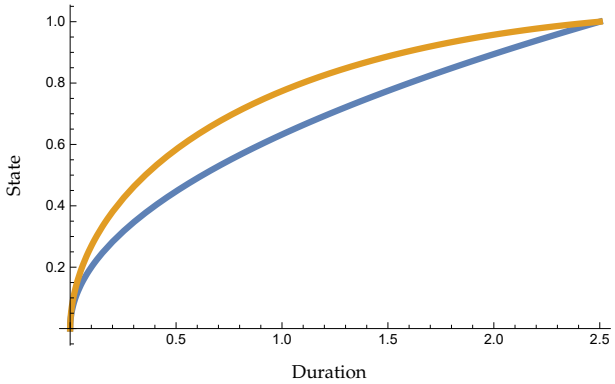

Figure 15. Strategy OPT (Blue) and online strategies for $t = 0.312$, $z = 1.1$ (Brown).

For progressive with a relatively small value of t, see Figures 16 and 17. As with earlier strategies, the competitive ratio is smaller when the online strategy changes to a lower state more aggressively. Turning to Figure 18, one observes that ratios are larger when states are changed more slowly. Referring to Figures 13 and 18, with a larger t value, progressive and offline strategy curves are rather congruent. This means that such an algorithm behaves more like the lower envelope. Next, larger values of z are analyzed.

Results are similar to what was observed before when t was increased. Ratios are worse (compare Figures 19 and 20), but the required increase to z values for a similar effect is greater. Increasing z makes the ratio go up at a more linear rate, compared to increases of t; in addition, the competitive ratio is somewhat lower when the value of t is increased. Either way, increasing t or z, the authors approach a strategy more as offline, and thereby to lower envelope behavior. Moreover, raising t effects a more rapid change in the schedule, and results in a larger and worse competitive ratio than increases to z, although there is a long period during the idle duration where the competitive ratio is better for higher t values. Significantly, ratios are larger for larger t values. In Table 6, competitive ratios for various such parameter values are given. We conclude that it is advantageous to use fast acceleration through smaller values of t and z. However, Table 6 reveals that things are not monotonic. For example, for $t = 0.4$ and $z = 0.2$, a competitive ratio of 1.67 results, whereas for the same z-value and $t = 0.32$ the competitive ratio is significantly larger with a value of 1.87. Thus, careful analysis is required for each individual system.

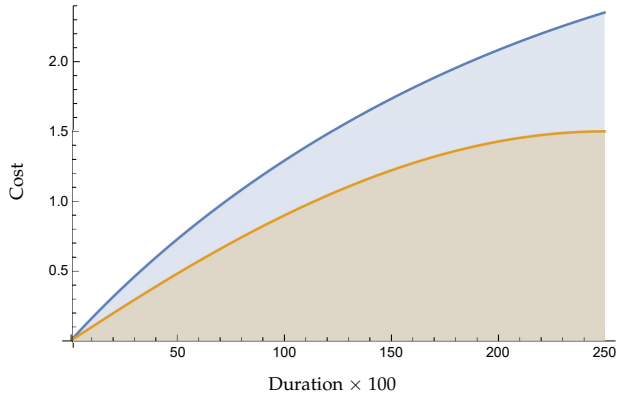

Figure 16. Costs Progressive$_{t,z}(r)$ for $t = 0.312$, $z = 0.1$ (Blue) and Cost OFFLINE (Brown).

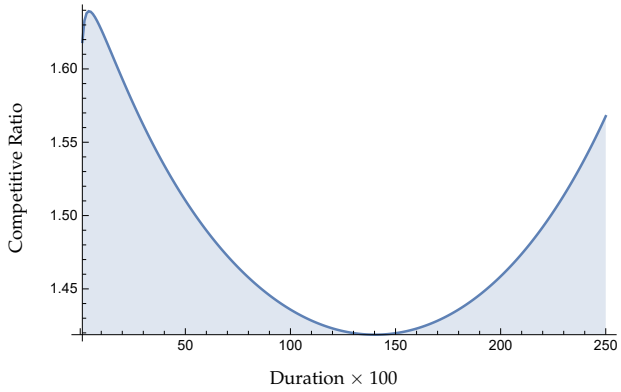

Figure 17. Competitive ratio $t = 0.312$, $z = 0.1$.

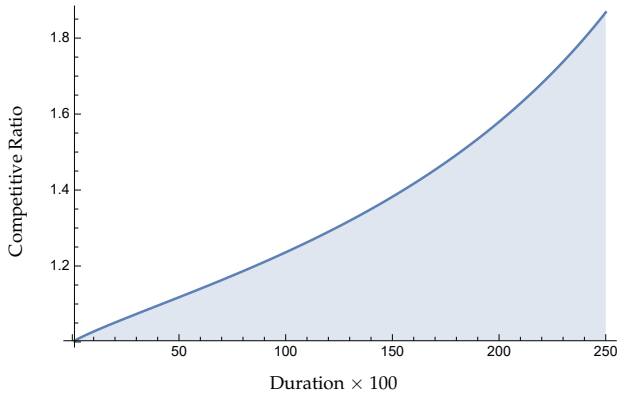

Figure 18. Competitive ratio $t = 0.712$, $z = 0.1$.

Table 6. Results for progressive$_{t,z}$ for various parameters. Competitive ratios are well below 2.

t Value	*z* Value	Competitive Ratio
0.32	0.2	1.87
0.32	0.1	1.58
0.39	0.1	1.63
0.50	0.1	1.73
0.40	0.2	1.67
0.60	0.2	1.81

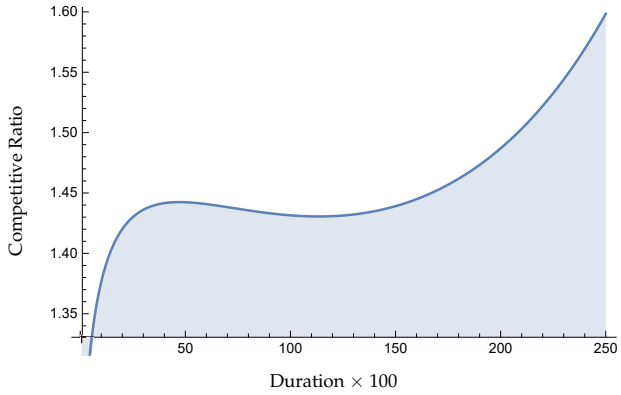

Figure 19. Competitive ratio $t = 0.312$, $z = 0.2$.

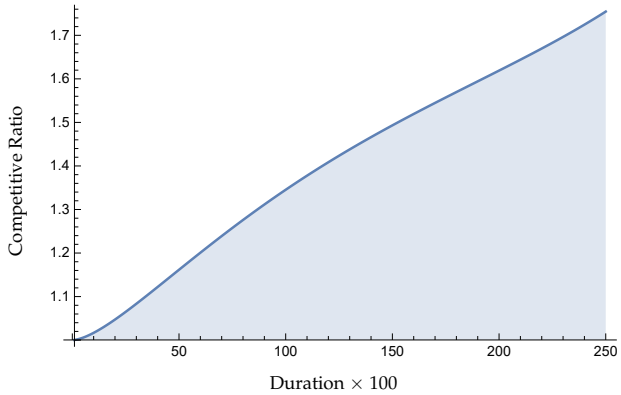

Figure 20. Competitive ratio $t = 0.312$, $z = 1.1$.

6. Conclusions

Power-down problems with few states, as well as power-down problems with one continuous state, play an important role in energy-efficiency for both information technology and the management of renewable energy in a truly resilient electrical grid. The authors have shown that in both situations—a few discrete states or one continuous state—competitive strategies exist and that those strategies can be efficiently found in closed form or through heuristics. One general rule that emerges from this work is that, in order to obtain good competitive ratios, the online strategy needs to switch to lower power states at a faster rate than for the offline strategy. This general insight can be useful for developing strategies for idle and power up cost types other than those considered in this paper.

The work by the authors shows that, even under the worst-case competitive analysis, ratios of significantly below the value of 2 can be achieved. This means that quantitative quality guarantees for a resilient smart grid were provided by this work. Table 5 shows that for systems with five states, a competitive ratio of close to 1.7 was achieved, i.e., significantly less than the best achievable value of 1.8 for three-states (see Table 3). For a continuous system, it is clear from our studies that progressive strategies generally achieve the best results, e.g., close to 1.6 in one case described in Table 6.

The authors suggest that such guarantees are crucial for policy makers as society transitions to a smart grid, which uses primarily renewable energy. A limitation of this study is that it does not address average case analysis, and it would be interesting in future work to perform simulation work to ascertain performance in practice for a smart grid as described in [1].

As mentioned in the Introduction, for discrete power down problems, Bein et al. [16,17] have introduced the decrease-and-reset technique, which decreases the standby time gradually when the frequency of the device usage becomes low. If we allow the worst-case competitive ratio to increase by only a small positive constant from the optimal value, we can design numerous algorithms other than

the optimal strategy that can be designed (which are nonetheless "near optimal"). Next, the authors introduce a parameter called "slackness degree", which represents the frequency of arrivals and can be constructed near optimal, which has superior competitiveness in the presence of slackness. Future work may consider applying decrease and reset in the continuous case as well.

Author Contributions: The three authors contributed equally to this paper. The order of authors is alphabetical.

Funding: The work of author Wolfgang Bein was supported by the National Science Foundation, Grant IIA 1427584. The work of author Hiro Ito was partially supported by CREST, JST, Grant No. JPMJCR1402 and KAKENHI, JSPS, Grant No. 15K11985.

Acknowledgments: The authors are grateful for the guidance given by Rüdiger Reischuk of Universität Lübeck during his sabbatical visit to the University of Nevada, Las Vegas, USA.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bein, W.; Madan, B.B.; Bein, D.; Nyknahad, D. Algorithmic Approaches for a Dependable Smart Grid. In *Information Technology: New Generations: 13th International Conference on Information Technology*; Latifi, S., Ed.; Springer: Berlin/Heidelberg, Germany, 2016; pp. 677–687.
2. Agarwal, Y.; Hodges, S.; Chandra, R.; Scott, J.; Bahl, P.; Gupta, R. Somniloquy: Augmenting Network Interfaces to Reduce PC Energy Usage. In Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, Boston, MA, USA, 22–24 April 2009; USENIX Association: Berkeley, CA, USA, 2009; pp. 365–380.
3. Agarwal, Y.; Savage, S.; Gupta, R. SleepServer: A Software-only Approach for Reducing the Energy Consumption of PCs Within Enterprise Environments. In Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference, Boston, MA, USA, 22–25 June 2010; USENIX Association: Berkeley, CA, USA, 2010; p. 22.
4. Nedevschi, S.; Chandrashekar, J.; Liu, J.; Nordman, B.; Ratnasamy, S.; Taft, N. Skilled in the Art of Being Idle: Reducing Energy Waste in Networked Systems. In Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation, Boston, MA, USA, 22–24 April 2009; USENIX Association: Berkeley, CA, USA, 2009; pp. 381–394.
5. Reich, J.; Goraczko, M.; Kansal, A.; Padhye, J. Sleepless in Seattle No Longer. In Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference, Boston, MA, USA, 22–25 June 2010; USENIX Association: Berkeley, CA, USA, 2010; p. 17.
6. Pruhs, K. Green computing algorithmics. In Proceedings of the 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS), Palm Springs, CA, USA, 22–25 October 2011; pp. 3–4.
7. Barroso, L. The price of performance. *ACM Queue* **2005**, *3*, 48–53. [CrossRef]
8. Albers, S. Energy-Efficient Algorithms. *Commun. ACM* **2010**, *53*, 86–96. [CrossRef]
9. Augustine, J.; Irani, S.; Swamy, C. Optimal Power-Down Strategies. In Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, Rome, Italy, 17–19 October 2004, pp. 530–539.
10. Augustine, J.; Irani, S.; Swamy, C. Optimal Power-Down Strategies. *SIAM J. Comput.* **2008**, *37*, 1499–1516. [CrossRef]
11. Borodin, A.; El-Yaniv, R. *Online Computation and Competitive Analysis*; Cambridge University Press: Cambridge, UK, 1998.
12. Hall, R.; Erdélyi, R.; Hanna, E.; Jones, J.M.; Scaife, A.A. Drivers of North Atlantic Polar Front jet stream variability. *Int. J. Climatol.* **2014**. [CrossRef]
13. Karlin, A.; Manasse, M.; Rudolph, L.; Sleator, D. Competitive snoopy caching. *Algorithmica* **1988**, *3*, 79–119. [CrossRef]
14. Irani, S.; Pruhs, K.R. Algorithmic Problems in Power Management. *SIGACT News* **2005**, *36*, 63–76. [CrossRef]
15. Karlin, A.R.; Kenyon, C.; Randall, D. Dynamic TCP Acknowledgement and Other Stories About e/(e-1). In Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing, Crete, Greece, 6–8 July 2001; ACM: New York, NY, USA, 2001; pp. 502–509.

16. Andro-Vasko, J.; Bein, W.; Nyknahad, D.; Ito, H. Evaluation of Online Power-Down Algorithms. In Proceedings of the 12th International Conference on Information Technology—New Generations, Las Vegas, NV, USA, 13–15 April 2015; pp. 473–478.
17. Bein, W.; Hatta, N.; Hernandez-Cons, N.; Ito, H.; Kasahara, S.; Kawahara, J. An Online Algorithm Optimally Self-tuning to Congestion for Power Management Problems. In Proceedings of the 9th International Conference on Approximation and Online Algorithms, Saarbrücken, Germany, 8–9 September 2011; pp. 35–48.
18. Bansal, N.; Chan, H.L.; Lam, T.W.; Lee, K.L. Scheduling for speed bounded processors. In Proceedings of the 35th International Colloquium on Automata, Languages and Programming, Reykjavik, Iceland, 7–11 July 2008; pp. 409–420.
19. Bansal, N.; Chan, H.L.; Pruhs, K.; Katz, D. Improved bounds for speed scaling in devices obeying the cube-root rule. In Proceedings of the 36th International Colloqium on Automata, Languages and Programming, Rhodes, Greece, 5–12 July 2009; pp. 144–155.
20. Han, X.; Lam, T.W.; Lee, L.K.; To, I.K.K.; Wong, P.W.H. Deadline scheduling and power management for speed bounded processors. *Theor. Comput. Sci.* **2010**, *411*, 3587–3600. [CrossRef]
21. Chen, J.J.; Kao, M.J.; Lee, D.; Rutter, I.; Wagner, D. Online dynamic power management with hard real-time guarantees. *Theor. Comput. Sci.* **2015**, *595*, 46–64. [CrossRef]
22. Albers, S.; Antoniadis, A. Race to Idle: New Algorithms for Speed Scaling with a Sleep State. In Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms, Kyoto, Japan, 17–19 January 2012; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 2012; pp. 1266–1285.
23. Anand, S.; Garg, N.; Megow, N. Meeting Deadlines: How Much Speed Suffices? In Proceedings of the ICALP 2011, Zurich, Switzerland, 4–8 July 2011.
24. Bein, W. Energy Saving in Data Centers. *Electronics* **2018**, *7*, 5. [CrossRef]
25. Dayarathna, M.; Wen, Y.; Fan, R. Data Center Energy Consumption Modeling: A Survey. *IEEE Commun. Surv. Tutor.* **2016**, *18*, 732–794. [CrossRef]
26. Borges, C.E.; Penya, Y.K.; Fernández, I. Evaluating Combined Load Forecasting in Large Power Systems and Smart Grids. *IEEE Trans. Ind. Inform.* **2013**, *9*, 1570–1577. [CrossRef]

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).